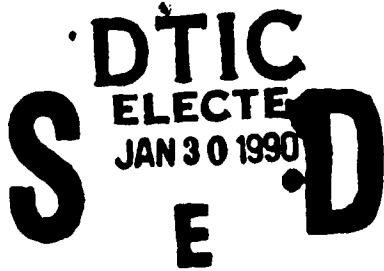


AD-A217 302

## REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE January 1990		3. REPORT TYPE AND DATES COVERED professional paper	
4. TITLE AND SUBTITLE FRACTAL-BASED IMAGE COMPRESSION				5. FUNDING NUMBERS PE: 0601152N WU: IC000036 PN: ZW10	
6. AUTHOR(S) S. M. Kocsis					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Ocean Systems Center San Diego, CA 92152-5000				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Director of Naval Laboratories Space and Naval Warfare Systems Command Washington, D.C. 20360				10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES					
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.				12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words)  Iterated Function Systems (IFS) offers a method of describing complicated digital files with a small set of functions exhibiting fractal properties. In coding an image, first cover it with contractive affine transformations of itself and then save the coefficients of the transformations. Decoding is performed by generating a dynamical system whose attractor is suitably close to the original image. The amount of distortion is dependent on the quality of the initial covering. This paper will describe the mathematics of IFS, the coding and decoding of a digital image with IFS, error analysis of IFS compression, and comparison to other compression techniques.					
					
Published in Conference <i>Proceedings</i> of the Twenty-Third Asilomar Conference on Signals, Systems, and Computers, October 1989.					
14. SUBJECT TERMS fractals feature classification image compression				15. NUMBER OF PAGES	
				16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UNLIMITED		

# Fractal-Based Image Compression

S. M. Kocsis

Naval Ocean Systems Center, Code 635, San Diego, CA 92152

## ABSTRACT

Iterated Function Systems (IFS) [1] offers a method of describing complicated digital files with a small set of functions exhibiting fractal properties. In coding an image, first cover it with contractive affine transformations of itself and then save the coefficients of the transformations. Decoding is performed by generating a dynamical system whose attractor is suitably close to the original image. The amount of distortion is dependent on the quality of the initial covering. This paper will describe the mathematics of IFS, the coding and decoding of a digital image with IFS, error analysis of IFS compression, and comparison to other compression techniques.

## 1 Introduction

Mandelbrot [3] coined the term *fractal* to describe sets that demonstrate detail under any arbitrary magnification. Mandelbrot described the properties of sets with non-integer dimension, *fractal sets*, and produced many examples from nature: *coastlines, clouds, brownian motion, trees, etc...* Human-made mechanisms also display fractal behavior: *the stock market, bouncing balls, dripping faucets, etc...* Mandelbrot [3] made it clear that in nature fractal action has a limit while pure mathematical fractals have structure at all scales, but both have in common *statistical invariance under transformation of scale*. An IFS generates a set with fractal properties. (20)

### 1.1 Properties of Fractal sets

Fractal sets are self-referencing, producing similar structure at smaller and smaller scales. This property causes many fractal sets to be continuous but non-differentiable. This property can be measured with the *fractal dimension*,  $D$ , a scalar that represents the limit of the number of discs of diameter  $\epsilon$ ,  $N(\epsilon)$ , to cover the set as  $\epsilon$  approaches zero.

$$D = \lim_{\epsilon \rightarrow 0} \frac{\ln N(\epsilon)}{\ln(1/\epsilon)}, \quad D = \text{fractal dimension} \quad (1)$$

### 1.2 Fractal Generation

One method of generating fractal sets is by starting with a seed pattern which is then scaled and repeated to produce the global set. Another method is iterative fractal construction which is the way that the IFS fractal is produced. An iterative fractal begins with the selection of an initial value  $X_0$ , which is the input to a function,  $f(X_n)$ . The resulting function value,  $X_1$ , is then used as the next input to the function with the outcome also applied as an input to the function. The structure of this mechanism is a feedback loop that generates a sequence of points,  $X_0, X_1, X_2, \dots, X_n$ , where  $X_{n+1} = f(X_n)$ . For this paper  $X_n = (x_n, y_n) \in \mathbb{R}^2$ . The overall result is a *Discrete-Dynamical System* (DDS) whose geometrical behavior is determined by the properties of the generating equations.

### 1.3 Dynamical Systems

A Dynamical System is a set of rules that govern the behavior of points in a well-defined space. Continuous dynamical systems can be transformed into discrete systems through application of the Poincaré map. We will only concern ourselves with the discrete case. An IFS is an example of a DDS. Upper-case,  $X_k$ , refers to points in  $\mathbb{R}^2$ . Lower case  $(x_k, y_k)$  refers to the coordinates of the point,  $X_k$ . We will look at sequences of points in  $\mathbb{R}^2$  that begin with an initial point and follow an orbit under the guidance of rules determined by a family of functions. The orbit is formed by the iteration of the functions.

### 1.4 Notation of DDS

The rule  $X_{k+1} = f(X_k)$  for generating a sequence of points,  $X_0, X_1, X_2, \dots, X_n \in \mathbb{R}^2$ , starts with the initial point  $X_0$ . The next points are:  $X_1 = f(X_0)$ ,  $X_2 = f(X_1)$ , ... and so on. The function  $f$  has 2 variables,  $x_k$  and  $y_k$ , which are the coordinates of the point  $X_k$ . This is the iteration process which takes the output of the function at each step

Accession For	
DTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	<input checked="" type="checkbox"/>

and feeds it back in as input. This feedback progress is what generates complex behavior from a set of simple rules. In function notation the iterates are:  $X_0, f(X_0), f(f(X_0)), f(f(f(X_0))), \dots$  OR  $X_0, f(X_0), f^2(X_0), f^3(X_0), \dots, f^n(X_0)$ . Both notations express the orbit of the original point,  $X_0$ . The latter, with superscripts, gives the number of iterates for  $X_0$  to arrive at another point in the sequence.

### 1.5 Definition of Attractor of a DDS

Of interest is the long-term behavior of the dynamical system. If the orbit of the dynamical system converges to some local neighborhood and once entering this neighborhood, stays for infinite time we say that the orbit has converged to an *Attractor*. Attractors are typically points, periodic orbits, or quasi-periodic cycles. If the attractor is none of the above, then it is termed a *Strange Attractor* which often has fractal properties.

**Definition 1 (Attractor)**  $P$  is an attractor of a DDS if there exists an open neighborhood of  $P$ ,  $U$ , such that if  $X \in U, X \neq P, \rightarrow \lim_{n \rightarrow \infty} f^n(X) = P$

### 1.6 Chaos in a DDS

DDS's with attractors possessing complex geometrical behavior are commonly *Strange Attractors* with fractal properties. Systems with strange attractors exhibit the properties of chaos. The IFS attractors are commonly strange attractors and are *Chaotic Systems* that arise from the iteration of deterministic sets of functions. Following the orbit of a DDS, once the sequence of points enters the boundary of the *Strange Attractor* it undergoes a transition from order to chaos. The orbit of a DDS once it enters the attractor is chaotic in two ways. If the initial point is perturbed even a small amount then the resulting orbit will eventually diverge from the original (*Sensitive Dependence on Initial Conditions*). Additionally the orbit within the attractor, given enough time, will come arbitrarily close to every interior point (*Topological Transitivity*).

## 2 Iterated Function Systems

For image compression an Iterated Function System (IFS) is constructed to have a unique attractor

that in most cases is chaotic with fractal properties. Hutchinson [2] and Barnsley [1] developed the concepts. An IFS is a set of contractive-affine transformations in  $\mathbf{R}^2$ . Associated with the IFS is a unique attractor, such that any random sequence of the IFS's affine-contractive mappings will converge to the attractor upon iteration in  $\mathbf{R}^2$ .

### 2.1 Definition of Affine Map

#### Definition 2 (Affine Map in $\mathbf{R}^2$ )

A Transformation  $W: \mathbf{R}^2 \rightarrow \mathbf{R}^2$ ,

$$W(X_k) = W \begin{pmatrix} x_k \\ y_k \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_k \\ y_k \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}$$

Affine maps in  $\mathbf{R}^2$  perform the operations of rotation, translation, and scaling on points  $X_k = (x_k, y_k) \in \mathbf{R}^2$ . The  $2 \times 2$  Matrix (a,b,c,d) accomplishes rotation and scaling. The  $2 \times 1$  Matrix (e,f) performs translation (a,b,c,d,e,f  $\in \mathbf{R}$ ).

### 2.2 Definition of Contractive Affine Map

A contractive affine map in  $\mathbf{R}^2$  brings points closer together. To define an IFS in  $\mathbf{R}^2$  we form a set of contractive-affine maps. The contractive property guarantees that the attractor will be well behaved and non-divergent. The IFS allows encoding a given image as a set of transformations. The original image can be discarded, its information represented as the coefficients of the affine transformations. This is the basis for IFS compression of digital images. To find the transformations that will produce a given image we apply Barnsley's *Collage Theorem* which describes a covering whose IFS will have a unique attractor suitably close to the original image.

#### Definition 3 (Contractive Affine Map in $\mathbf{R}^2$ )

Let  $d(X, Y)$  be a distance function defined for all  $X, Y$  in  $\mathbf{R}^2$ . Let  $s$  be  $0 \leq s < 1$ ,  $s = \text{Contractive factor}$ , THEN a contractive affine map  $W_i: \mathbf{R}^2 \rightarrow \mathbf{R}^2$ , satisfies  $d(W_i(X), W_i(Y)) \leq sd(X, Y)$  for all  $X, Y$  in  $\mathbf{R}^2$ .

### 2.3 Definition of IFS

An IFS is defined as a finite set of contractive affine-mappings in  $\mathbf{R}^2$  that map the space into itself. The contractivity factor of an IFS is the maximum

of the individual contractive factors of the mappings. This definition guarantees that each IFS has a unique attractor. However, it is not the case that each attractor has a unique IFS.

## 2.4 Collage Theorem

The *Collage Theorem* [1] says that given an arbitrary image defined in a closed bounded space, such as a computer screen, we can construct an IFS with a unique attractor, that we can force to be arbitrarily close to the original image.

**Theorem 1 (Barnsley '85)** Let  $\{W_i : W_i \in \mathbb{R}^2, i = 1, \dots, n\} = \text{IFS}$ . Let  $L \subset \mathbb{R}^2$ , with  $d(L \cup W_i(L)) \leq \epsilon$ ,  $\epsilon > 0$ , THEN  $d(L, A) \leq \epsilon / (1 - s)$ , Where  $A = \text{Attractor of IFS}$ ,  $s = \text{Contractivity Factor}$ .

The *Collage Theorem* [1] states that the inverse problem can be solved. One method is to cover an image with contractive-affine mappings of itself. Then solve the transformations to obtain the IFS. Once the IFS is constructed the attractor is generated by assigning a probability weighting to each map then following a random orbit that will eventually converge to the attractor. If the covering was done with care it will be suitably close to the original image.

## 2.5 Collage Theorem algorithm

For fast image compression the best situation would be an automatic algorithm for generating the IFS of an arbitrary digital image. Unfortunately it is difficult to implement such a process because of the numerous degrees of freedom possible in covering with the affine transformations. The example in this paper was done by tiling the image with contracted copies of itself controlled by affine maps. The covering was done by issuing commands from the keyboard and mouse that applied the affine maps on the computer screen. Once the ensemble of maps adequately covered the image to be compressed, they were saved in a file to be used in generating the attractor.

# 3 IFS Attractor

Each IFS has a unique attractor, but the reverse is not true. Given an IFS, its attractor can be found by a random iteration starting with the initial

point. The beginning iterations may be discarded since they represent a transient containing no information. Once the sequence of iterations enters the attractor it will never leave.

## 3.1 Random Algorithm for IFS Attractor

The random algorithm starts with an arbitrary initial point  $X_0 = (x_0, y_0) \in \mathbb{R}^2$ . One of the mappings of the IFS,  $W_k$ , is then selected at random and the next point,  $X_1$ , is found by evaluating  $W_k(X_0) = X_1$ . Hence,  $X_2 = W_m(X_1)$  with  $W_m$  picked at random. Successive points are generated similarly.

Let  $\{W_i : W_i \in \mathbb{R}^2, i = 1, \dots, n\} = \text{IFS}$ ,  $\forall W_i, (i = 1, \dots, n), \exists P_i - \sum_{i=1}^n P_i = 1$  (see sec. 3.4) Now define a sequence of points  $\{X_i\}_{i=1}^n \rightarrow X_n = W_k(X_{n-1})$  with the constraint  $P(W_k) = P_k$ . This algorithm generates a discrete dynamical system  $X_0, X_1, X_2, \dots, X_n$  that converges to a unique attractor. Each point in the sequence is generated by one of the maps picked at random. As long as the initial point is within the IFS's basin of attraction then the DDS will eventually converge to the attractor. The number of iterations for the random sequence to converge to the attractor depends on the distance of the initial point from the boundary of the IFS's attractor. Generally the first 50 iterations represents a transient and should be discarded.

## 3.2 Deterministic Algorithm for IFS Attractor

An alternate method of finding the *attractor* of an IFS is to simultaneously apply all the mappings, of the IFS, to an initial set of points rather than just one point. In this algorithm no probabilities are assigned to each map causing all the functions to contribute to the attractor equally. This method is slower than the random algorithm, unless programmed on a parallel processor, and has less versatility in producing different textures in the interior of the attractor, since the distribution of the mappings is uniform (see sec. 3.4).

## 3.3 Finding the IFS

To apply the collage theorem in  $\mathbb{R}^2$ , cover the target image with contractive-affine copies of itself then solve for the coefficients that map the original image to each of the tiles. For example consider the affine function necessary to map a large triangle,  $T$ , to a contractive copy of itself,  $t$ . Identify 3 tie points on

T and the 3 corresponding tie points on t. Each point of T,  $(x_k, y_k)$ , is mapped to a point of t,  $(x'_k, y'_k)$  ( $k = 1, 2, 3$ ), by the contractive-affine mappings:

$$W \begin{pmatrix} x_k \\ y_k \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_k \\ y_k \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} = \begin{pmatrix} x'_k \\ y'_k \end{pmatrix} \quad (2)$$

With 3 tie points the affine map can be represented as 6 linear equations, which can be partitioned into 2 systems of simultaneous linear equations, of 3 equations in 3 unknowns, sufficient to solve for the coefficients:  $a, b, c, d, e$ , and  $f$ .

Solve for  $a, b$ , and  $e$  as a system of linear equations.

$$\begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ e \end{pmatrix} = \begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} \quad (3)$$

Solve for  $c, d$ , and  $f$  similarly.

$$\begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{pmatrix} \begin{pmatrix} c \\ d \\ f \end{pmatrix} = \begin{pmatrix} y'_1 \\ y'_2 \\ y'_3 \end{pmatrix} \quad (4)$$

Each mapping contained in the IFS for  $\mathbf{R}^2$  is represented by 6 numbers:  $a, b, c$ , and  $d$  which denote rotation and scaling, while  $e$  and  $f$  signify translation. If there are  $n$  mappings then the IFS is encoded with  $6n$  coefficients. To generate the attractor of the IFS, equivalent to decoding, a probability weighting is assigned to each map.

### 3.4 Assigning probabilities to IFS

Suppose there are  $n$  mappings in the IFS, then to use the random algorithm to generate the attractor it is necessary to assign  $n$  probability weightings, one to each map. The sum of the probabilities should be one.

Determine the probability,  $P_i$ , of each  $W_i$  by taking the absolute value of the determinant of the rotation-scaling matrix. This weighting is most accurate if the coverings do not overlap. The sum is normalized to unity by dividing by the absolute sum of all the probability weightings. This insures that each map is randomly selected, directly proportional to the information it contributes to the attractor.

$$P_i = \frac{\left| \det \begin{pmatrix} a_i & b_i \\ c_i & d_i \end{pmatrix} \right|}{\sum_{i=1}^n \left| \det \begin{pmatrix} a_i & b_i \\ c_i & d_i \end{pmatrix} \right|} = \frac{|a_i d_i - b_i c_i|}{\sum_{i=1}^n |a_i d_i - b_i c_i|} \quad (5)$$

### 3.5 IFS Probability Distribution

Assigning a probability to each map of the IFS determines the distribution of points in the interior of the attractor. Lowering the probability of a map lowers the frequency of points visited by that map. Changing the probabilities does not change the boundaries of the attractor, what changes is the distribution of the attractor's interior points. The attractor's boundaries remain invariant as long as the affine-map coefficients remain the same. By counting the total number of iterations that visit the attractor and the number that visit each pixel a histogram of the DDS density is generated. The relative density,  $rdense$ , of each pixel would be:  $rdense = N(i, j) / Iter$ , with  $N(i, j)$  = total iterations at pixel location  $(i, j)$  and  $Iter$  = total number of iterations inside the attractor. Once  $rdense$  is computed a color lookup table could be constructed to assign color values to the attractor based on the histogram. Textures could also be generated by reassigning the probabilities to generate a different histogram with corresponding density.

## 4 Measurements of IFS

To measure the error between the original image,  $L$ , and the IFS's attractor,  $A$ , two methods are used in this paper. The first is relative difference  $rdiff$  and the second is the Hausdorff distance  $hd$  between  $L$  and  $A$ . Both methods assume monochrome digital images. The  $rdiff$  measures points present in  $L$  but not in  $A$  and is dimensionless. The  $rdiff$  measure would be 0 for an errorless compression, 1 for the worst-possible case.

$$rdiff = \left( \frac{\text{number of points in } |L - A|}{\text{total number of points in } L} \right)^{1/2} \quad (6)$$

The Hausdorff distance is a metric between 2 sets in a metric space, in our case the space  $\mathbf{R}^2$ . The Hausdorff metric is found by first finding the minimum of the distance from all the points in  $L$  to the points of  $A$ . Next the minimum of the distance from all the points in  $A$  to the points of  $L$  is computed. The maximum of these 2 minimums is the Hausdorff distance. Because the metric space the algorithm works in is a computer screen, the  $hd$  units are in pixels.

$$hd(L, A) = \text{Max} (\text{Min}(d(L, A)), \text{Min}(d(A, L))),$$

Where  $d$  = distance function

#### 4.1 IFS Compression

A computer experiment was conducted with the target image being a digitized map of the U.S.A.. The *Collage Theorem* was applied and 50 contractive affine transformations constructed that covered the original image. Associated probability weightings were computed, as in sec. 3.4. The random algorithm was used to generate the attractor of the IFS. The compression results are measured in 2 ways:

$$ratnum = \frac{\text{total of numbers to describe L}}{\text{total of numbers to describe A}} \quad (7)$$

$$ratbyt = \frac{\text{number of bytes in L}}{\text{number of bytes in A}} \quad (8)$$

For example, if L could be described with 1500 (x,y) coordinates and A's IFS had 50 affine mappings then  $ratnum = (200 * 2)/(6 * 50) = 10$ . Additionally if L is 5000 bytes and A is 1000 bytes then  $ratbyt = 5000/1000 = 5$ .

#### 4.2 Results of IFS Computer Experiment

As a comparison the commonly used Ziv-Lempel-Welch compression algorithm (ZLW) [5] [4] was used on the same image.

Compression	IFS	ZLW
ratnum	10/1	N.A.
ratbyt	100/1	5/1
rdiff	0.099	0.0
hd	6.4 pixels	0.0 pixels

For this image, IFS supplies better compression than ZLW but not without a loss in image fidelity. The ZLW algorithm gives lossless compression while the IFS compression introduces distortion in translation, scaling, and rotation. To decrease the error of the IFS compression more transforms need to be used which would decrease the compression ratio. Each transform contributes 6 coefficients to the IFS code. The ideal IFS optimization would minimize the error and with the minimum number of maps.

## 5 Summary

Compression with IFS is accomplished by finding a set of contractive-affine maps that cover the orig-

inal set, an application of Barnsley's *Collage Theorem*. The 6 parameters per map are saved as the code for the compression. To reconstruct the original image, a probability weighting is assigned to each map. The weighting is determined by the information each map contributes to the attractor and is computed strictly from the parameters. To generate the attractor an initial point is selected which is used as input to one of the maps selected at random. All the successive points are found by iteration of the maps selected by random. This generates a random sequence of points that, after settling down, converges to the attractor. The attractor of this discrete-dynamical system will be within epsilon distance of the original image. Epsilon is determined by the quality of the covering. This compression method works best with images that display self-similarity, such as coast lines.

## 6 Acknowledgments

This work was performed with the support of the N.O.S.C. Independent Research program which is directed by Dr. Alan Gordon. The author wishes to thank Hal Arnold for help in designing the computer experiments and for software development. Additional thanks to Roger Boss, Bill Jacobs, Joe Fitchek, Martin Jordan, and Hendrik VanBrundt for technical and software assistance.

## References

- [1] M. Barnsley, *Fractals Everywhere*, Academic Press, San Diego, 1988
- [2] J. Hutchinson, *Fractals and Self Similarity*, Indiana University Journal of Mathematics, 30:713-747 (1981)
- [3] B. Mandelbrot, *The Fractal Geometry of Nature*, W.H. Freeman and Co., San Francisco, 1982
- [4] Terry A. Welch, *A Technique for High Performance Data Compression*, IEEE Computer, vol. 17, no. 6 (June 84)
- [5] J. Ziv and A. Lempel, *Compression of Individual Sequences via Variable-Rate Coding*, IEEE Trans. Inform. Theory IT-24, 5 (Sept. 1978), 530-536.